

AD \_\_\_\_\_

Award Number: DAMD17-98-C-8035

TITLE: Actigraph and Sleep Data Analysis

PRINCIPAL INVESTIGATOR: Russell C. Eberhart, Ph.D.

CONTRACTING ORGANIZATION: Advancetek, Incorporated  
Indianapolis, Indiana 46202

REPORT DATE: April 1999

TYPE OF REPORT: Final

PREPARED FOR: U.S. Army Medical Research and Materiel Command  
Fort Detrick, Maryland 21702-5012

DISTRIBUTION STATEMENT: Approved for Public Release;  
Distribution Unlimited

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision unless so designated by other documentation.

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE April 1999	3. REPORT TYPE AND DATES COVERED Final (15 May 98 - 14 Feb 99)	
4. TITLE AND SUBTITLE Actigraph and Sleep Data Analysis			5. FUNDING NUMBERS DAMD17-98-C-8035	
6. AUTHOR(S) Russell C. Eberhart, Ph.D.				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Advancetek, Incorporated Indianapolis, Indiana 46202			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Medical Research and Materiel Command Fort Detrick, Maryland 21702-5012			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES			19990820 026	
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)				
14. SUBJECT TERMS			15. NUMBER OF PAGES 17	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited	

## FOREWORD

Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the U.S. Army.

\_\_\_\_ Where copyrighted material is quoted, permission has been obtained to use such material.

\_\_\_\_ Where material from documents designated for limited distribution is quoted, permission has been obtained to use the material.

\_\_\_\_ Citations of commercial organizations and trade names in this report do not constitute an official Department of Army endorsement or approval of the products or services of these organizations.

\_\_\_\_ In conducting research using animals, the investigator(s) adhered to the "Guide for the Care and Use of Laboratory Animals," prepared by the Committee on Care and use of Laboratory Animals of the Institute of Laboratory Resources, national Research Council (NIH Publication No. 86-23, Revised 1985).

\_\_\_\_ For the protection of human subjects, the investigator(s) adhered to policies of applicable Federal Law 45 CFR 46.

\_\_\_\_ In conducting research utilizing recombinant DNA technology, the investigator(s) adhered to current guidelines promulgated by the National Institutes of Health.

\_\_\_\_ In the conduct of research utilizing recombinant DNA, the investigator(s) adhered to the NIH Guidelines for Research Involving Recombinant DNA Molecules.

\_\_\_\_ In the conduct of research involving hazardous organisms, the investigator(s) adhered to the CDC-NIH Guide for Biosafety in Microbiological and Biomedical Laboratories.

---

PI - Signature

Date

**ACTIGRAPHY PROJECT  
WALTER REED ARMY INSTITUTE OF RESEARCH**

**Submitted by**

**Biomedical Engineering Center  
Purdue School of Engineering and Technology, IUPUI**

### **Introduction**

During the third quarter of activity, Purdue School of Engineering and Technology representatives continued to process data files received from WRAIR, participated in two project meetings, one at WRAIR in Washington, DC, and one in Indianapolis, and worked with WRAIR representatives to refine the project objective. The system design deliverable was completed and is included with this report. Some support activities for WRAIR's upcoming "Rosetta Stone" study were also carried out. Following are summaries of each of these activities. Working on the project during this time period were Prof. Russ Eberhart, Principal Investigator, and Mr. Xiaohui Hu, Research Assistant.

### **Project Meetings**

Project meetings were held at WRAIR in March 1999 and in Indianapolis in April 1999. Methods of data analysis were discussed. At the meeting in April, it was decided to further refine the work done in phase I to include a software tool to predict PVT performance (reaction times) based on polysomnography data (sleep scoring). The system design information presented at the April meeting was deemed sufficient to meet the deliverable requirement, for which an extension until May 14, 1999 has been requested.

### **System Design Deliverable**

The primary goal of Phase I of this contract is to develop a software analysis tool that classifies a subject as asleep or awake from actigraph data. Neural network tools that achieve a testing accuracy of 93-95 percent have been developed using particle swarm optimization. Included with this quarterly report is a diskette with the trained neural network weights that achieved these results (files r201.wts and r202.wts) and the particle swarm based neural network executable code (file psonn99.exe). Also included is the Visual Basic demonstration code for particle swarm optimization (file swarm98a.zip). The demonstration code can be used to optimize several benchmark functions, which are included.

Attached as Appendix A is a printout of training and test results that includes results for the two above-mentioned weight files. It can be seen that the performance on test data (not used for training) was about 95.6% correct for r201.wts and 93.2% for r202.wts. Attached as Appendix B is a listing of the Visual Basic source code for the particle swarm demonstration.

### **Data Preprocessing**

As of May 6, 1999, we have received 65 actigraph data sets on CD. A listing of these data sets is attached as Appendix C. Also listed in Appendix C are the four sleep data files we have received.

Each sleep data file starts with the sleep period of day 3 at 23:00. Seven days of sleep of a specified length (3, 5, 7, or 9 hours) follows, followed by three days of recovery (9 hours per night).

During the April project meeting, the preprocessing of the actigraph and sleep files was reviewed in detail. A flowchart describing the preprocessing is attached as Appendix D. During the discussion of the flowchart, it was pointed out that actigraph data values are for the *previous* four seconds, where sleep scores are for the *next* 30 seconds (or 16 seconds during SLT). It was decided to shift all actigraph data 4 seconds "to the left" (earlier) so that it is consistent with the sleep scores.

One item was not resolved. When we process the data into 60-second blocks, there are occurrences of "1 0" or "0 1" (sleep wake or wake sleep). We need to know whether to score these one-minute periods as sleep or wake. WRAIR is supposed to get an answer to us very soon.

### **Data Analysis**

During the April project meeting, it was decided that we would focus on prediction of PVT data using the sleep-wake scoring and the WRAIR performance model. We will use particle swarm optimization to evolve the performance model parameters.

WRAIR will provide the performance model as soon as a confidentiality agreement in the form of a CRADA is executed. IUPUI signed the CRADA on or about April 20, 1999, and forwarded it to WRAIR. We have not received the executed CRADA as of May 6, 1999. Receipt of the WRAIR performance model is necessary for us to proceed.

### **TeleActigraph-Related Activities**

Activities related to Precision Control Design's TeleActigraph (TAG) system were carried out during the quarter that were designed to support WRAIR's upcoming "Rosetta Stone" study. These activities, which were coordinated with Precision Control Design staff, included:

- 1) Writing and compiling a program to pre-process raw data as downloaded from the belt unit into an ASCII data file with one line per reading. This program was provided to WRAIR in February 1999.
- 2) Writing and revising a Matlab script to load a data file and display various attributes including power spectral density, and normalized amplitude versus frequency. This script was provided to WRAIR in February 1999.
- 3) Acquiring resting (awake) data from five additional "normal" subjects, from which amplitude and frequency characteristics of typical wrist tremor (with unsupported elbow) can be derived. We now have data on 12 normal subjects. This data will be useful for the Rosetta Stone study.
- 4) Becoming more familiar with TAG System operation so as to provide observations and guidance to WRAIR on system operation during the Rosetta Stone study. Observations and guidance (in addition to that in that last Quarterly Report) include:
  - a) Real-time data acquisition (instead of off-line downloading of data) still seems to be a very good way to acquire data from the belt unit, at least during sleep periods. We have ordered a 12-bit National digital-to-analog PCMCIA card for use in a laptop computer that is running LabView for Windows. A LabView script has been written that permits monitoring of data as it is acquired and stored, which has several obvious advantages.
  - b) We have installed a vibration testing system featuring a VTS600 600-watt amplifier and a Model VG 100 vibrator (Vibration Test Systems, Aurora, Ohio). It should be noted that this system was received "on loan" from Dr. David Burr at the Indiana University Medical School at no cost to this project. We use a Techtronics waveform generator or a pulse generator to drive the system, and monitor results with a digital scope and a PC with Labview installed.

TAG units have been preliminarily characterized and calibrated, and benchmarked against the BMA-32 actigraph units that will be used concurrently in the Rosetta Stone study. Results of this work were provided to WRAIR and PCD staff members at the April project meeting. It appears that one count from the TAG unit in Mode 9 corresponds to an acceleration of about 2 milli-gravities. Further work will be documented in the next quarterly report.

- c) Some concern exists with respect to the fidelity of data being received from the TAG units. Driving them with sine waves with less than one percent harmonic distortion results in output data that is not as nearly a sine wave as might be desired. Output data examples obtained when driving the TAG with a 6mm peak-peak sine wave at 1, 2, and 3 Hz are attached as Appendix E. This matter is being investigated with the full cooperation of PCD. We are also examining our test configuration. Nothing is being ruled out at this time.

Date: 1999-03-10 14:09:58

Neural Net topology: 7 inputs, 14 Hiddens, 2 outputs

Training is stopped: 14000 iterations or 0.025000 errors

TrainSet: 157200.tr

TestSet : 157200.te

	class00	class01	total
Weights : r200.WTS			
TrainSet:	0.875519 ( 211/ 241)	0.911950 ( 145/ 159)	0.890000 ( 356/ 400)
Rules :	0.879668 ( 212/ 241)	0.937107 ( 149/ 159)	0.902500 ( 361/ 400)
TestSet :	0.895323 ( 402/ 449)	0.769231 ( 60/ 78)	0.876660 ( 462/ 527)
Rules :	0.917595 ( 412/ 449)	0.794872 ( 62/ 78)	0.899431 ( 474/ 527)
Weights : r201.WTS			
TrainSet:	0.937759 ( 226/ 241)	0.918239 ( 146/ 159)	0.930000 ( 372/ 400)
Rules :	0.921162 ( 222/ 241)	0.930818 ( 148/ 159)	0.925000 ( 370/ 400)
TestSet :	0.959911 ( 431/ 449)	0.820513 ( 64/ 78)	0.939279 ( 495/ 527)
Rules :	0.977728 ( 439/ 449)	0.833333 ( 65/ 78)	0.956357 ( 504/ 527)
Weights : r202.WTS			
TrainSet:	0.904564 ( 218/ 241)	0.962264 ( 153/ 159)	0.927500 ( 371/ 400)
Rules :	0.908714 ( 219/ 241)	0.981132 ( 156/ 159)	0.937500 ( 375/ 400)
TestSet :	0.917595 ( 412/ 449)	0.884615 ( 69/ 78)	0.912713 ( 481/ 527)
Rules :	0.937639 ( 421/ 449)	0.897436 ( 70/ 78)	0.931689 ( 491/ 527)

Date: 1999-03-10 14:09:55

Neural Net topology: 7 inputs, 14 Hiddens, 2 outputs

Training is stopped: 16000 iterations or 0.025000 errors

TrainSet: 157200.tr

TestSet : 157200.te

	class00	class01	total
Weights : r21a0.WTS			
TrainSet:	0.937759 ( 226/ 241)	0.886792 ( 141/ 159)	0.917500 ( 367/ 400)
Rules :	0.950207 ( 229/ 241)	0.918239 ( 146/ 159)	0.937500 ( 375/ 400)
TestSet :	0.955457 ( 429/ 449)	0.717949 ( 56/ 78)	0.920304 ( 485/ 527)
Rules :	0.979955 ( 440/ 449)	0.743590 ( 58/ 78)	0.944972 ( 498/ 527)
Weights : r21a1.WTS			
TrainSet:	0.941909 ( 227/ 241)	0.886792 ( 141/ 159)	0.920000 ( 368/ 400)
Rules :	0.921162 ( 222/ 241)	0.899371 ( 143/ 159)	0.912500 ( 365/ 400)
TestSet :	0.937639 ( 421/ 449)	0.730769 ( 57/ 78)	0.907021 ( 478/ 527)
Rules :	0.962138 ( 432/ 449)	0.769231 ( 60/ 78)	0.933586 ( 492/ 527)

' Particle swarm VB5 software by Russ Eberhart  
' Make sure that the Open statement on line 40 points to the correct directory  
' May 18, 1998, graph iGbest, iLbest  
' May 29, 1998, added f0, f1, f2, f3, f5

Option Explicit

Option Base 1

Dim iPOPSIZE As Integer	' Population size
Dim iDIMENSIONS As Integer	' Number of dimensions
Dim iPopindex As Integer	' Index for population
Dim iDimindex As Integer	' Index for dimensions
Dim fINITWT As Single	' Initial inertia weight
Dim fMAXVEL As Single	' Maximum velocity allowed
Dim nMAXITER As Integer	' Maximum number of iterations
Dim nIter As Integer	' Number of iterations
Dim fPos() As Single	' Position for each particle
Dim fTempPos() As Single	' Temporary updated position for each particle
Dim fVel() As Single	' Velocity for each particle
Dim fDumVel() As Single	' Dummy velocity vector 1-dim array
Dim fBestPos() As Single	' Best previous position for each particle
Dim fMaxPos As Single	' Dynamic range
Dim fInerWt As Single	' Inertia weight
Dim fErrVal() As Double	' Function error value calculated by eval()
Dim fPbestVal() As Double	' Best error value over time for each particle
Dim fERRCUTOFF As Double	' Error value at which system stops
Dim iUSEBETTER As Integer	' Usebetter is kind of a momentum
Dim iBetter() As Integer	' This gets set in program
Dim iGbest As Integer	' Index for global best particle
Dim iLbest As Integer	' Index for local (neighborhood) best particle
Dim iLOCAL As Integer	' Neighborhood size specified in run file
Dim iHOODSIZE As Integer	' Neighborhood size used in program
Dim iHOODINDEX As Integer	' Neighborhood index (offset from particle)
Dim iNeighbor() As Integer	' Popindexes of neighbors resulting from iLOCAL
Dim sOutfile As String	' Filename for output
Dim iXcoord As Integer, iYcoord As Integer	' x & y coords for plots
Dim iFNCTNO As Integer	' Function number to be optimized (1=f1, etc.)
Dim fErrValDim As Double	' Error value for a single dimension
Const conPI = 3.14159265358979	' Define Pi

Private Sub cmdSwarm\_Click()

Open "c:\mydocs\vb\psinput.txt" For Input As #1  
Input #1, iPOPSIZE, iFNCTNO, iDIMENSIONS, fERRCUTOFF, fMAXVEL, fMaxPos, \_  
nMAXITER, iUSEBETTER, fINITWT, iLOCAL, sOutfile  
' Population size, function no. to be optimized, no. of dimensions,  
' error cutoff value, maximum velocity, maximum position  
' (range), maximum no. of iterations, use momentum (usebetter), initial  
' inertia weight, neighborhood (0=global, even no. for local neighborhood),  
' filename for output  
Close #1  
  
If iFNCTNO = 6 Or iFNCTNO = 5 Then iDIMENSIONS = 2 'Set # of dimensions for f5 & f6 fnctns  
  
ReDim fPos(iPOPSIZE, iDIMENSIONS)  
ReDim fTempPos(iPOPSIZE, iDIMENSIONS)  
ReDim fVel(iPOPSIZE, iDIMENSIONS)  
ReDim fBestPos(iPOPSIZE, iDIMENSIONS)  
ReDim fDumVel(iDIMENSIONS)  
ReDim fErrVal(iPOPSIZE)  
ReDim fPbestVal(iPOPSIZE)  
ReDim iBetter(iPOPSIZE)  
ReDim iNeighbor(-iPOPSIZE To iPOPSIZE)  
  
picSwarm1.BackColor = vbBlack  
  
If iLOCAL > 0 Then iHOODSIZE = Int(iLOCAL / 2#) \* 2# Else iHOODSIZE = iPOPSIZE  
  
Randomize  
' Randomize the positions and velocities for entire population



```

For iPopindex = 1 To iPOPSIZE
  For iDimindex = 1 To iDIMENSIONS
    fPos(iPopindex, iDimindex) = Rnd * fMaxPos
    fBestPos(iPopindex, iDimindex) = fPos(iPopindex, iDimindex)
    fVel(iPopindex, iDimindex) = Rnd * fMAXVEL
    If Rnd > 0.5 Then fPos(iPopindex, iDimindex) = -fPos(iPopindex, iDimindex)
    If Rnd > 0.5 Then fVel(iPopindex, iDimindex) = -fVel(iPopindex, iDimindex)
    'frmSwarm.Print "Init. Pos & Vel ", fPos(iPopindex, iDimindex) _
    , fVel(iPopindex, iDimindex) ' for diagnostic purpose
  Next iDimindex
Next iPopindex

' Main swarm loop here
For nIter = 1 To nMAXITER

  ' Update inertia weight; linear from fINITWT to 0.4
  fInerWt = ((fINITWT - 0.4) * (nMAXITER - nIter) / nMAXITER) + 0.4

  For iPopindex = 1 To iPOPSIZE 'MAIN main loop starts here
    'Setup dummy velocity vector for current population member
    For iDimindex = 1 To iDIMENSIONS
      fDumVel(iDimindex) = fVel(iPopindex, iDimindex)
    Next iDimindex

    iBetter(iPopindex) = 0 ' Set to 0 unless new Pbest achieved

    Select Case iFNCTNO ' returns fErrVal(iPopindex)
      Case 0
        Call evalf0(iPopindex) ' evals f0 spherical function error
      Case 1
        Call evalf1(iPopindex) ' evals f1 Rosenbrock function error
      Case 2
        Call evalf2(iPopindex) ' evals f2 Rastrigrin function error
      Case 3
        Call evalf3(iPopindex) ' evals f3 Griewank function error
      Case 5
        Call evalf5(iPopindex) ' evals f5 He function error
      Case 6
        Call evalf6(iPopindex) ' evals f6 Schaffer function error
    End Select

    If nIter = 1 Then
      fPbestVal(iPopindex) = fErrVal(iPopindex)
      iGbest = 1
    End If

    If fErrVal(iPopindex) < fPbestVal(iPopindex) Then 'If new Pbest
      fPbestVal(iPopindex) = fErrVal(iPopindex)

      For iDimindex = 1 To iDIMENSIONS ' Reset Pbest location vector
        fBestPos(iPopindex, iDimindex) = fPos(iPopindex, iDimindex)
      Next iDimindex

      If fPbestVal(iPopindex) < fPbestVal(iGbest) Then
        ' color(iGbest) = DEFAULTCOLOR need to dim these variables
        iGbest = iPopindex
        ' color(iGbest) = GBESTCOLOR
      End If

      If iUSEBETTER = 1 Then iBetter(iPopindex) = 1
    End If ' End new Pbest loop
  Next iPopindex 'end MAIN main loop for gold gbest only

  'update velocity, position, graph position
  ' Does neighborhood calculation of iLbest

```

```

If iLOCAL > 0 Then
  For iHOODINDEX = 0 To iHOODSIZE
    iNeighbor(iHOODINDEX) = iPopindex - (iHOODSIZE / 2#) + iHOODINDEX
    'for iPopindex = 1, goes from 0 to 2 for iHOODSIZE of 2
    'from -1 to 3 for iHOODSIZE of 4
    ' Now wrap the ends of the array
    If iNeighbor(iHOODINDEX) < 1 Then iNeighbor(iHOODINDEX) = iPOPSIZE + _
      iNeighbor(iHOODINDEX)
    If iNeighbor(iHOODINDEX) > iPOPSIZE Then iNeighbor(iHOODINDEX) = _
      iNeighbor(iHOODINDEX) - iPOPSIZE
    'Start with iNeighbor(0) as iLbest and try to beat it
    If iHOODINDEX = 0 Then iLbest = iNeighbor(0)
    If fPbestVal(iNeighbor(iHOODINDEX)) < fPbestVal(iLbest) Then _
      iLbest = iNeighbor(iHOODINDEX)
  Next iHOODINDEX
End If

If iLOCAL = 0 Then iLbest = iGbest

' Update velocity vector for one particle Russ Reduced version
For iDimindex = 1 To iDIMENSIONS 'fInerWt below
  fVel(iPopindex, iDimindex) = (0.5 + (Rnd / 2)) * fVel(iPopindex, iDimindex) + _
    2# * Rnd * (fBestPos(iPopindex, iDimindex) - fPos(iPopindex, iDimindex)) + _
    2# * Rnd * (fBestPos(iLbest, iDimindex) - fPos(iPopindex, iDimindex))

  If fVel(iPopindex, iDimindex) > fMAXVEL Then
    fVel(iPopindex, iDimindex) = fMAXVEL
  ElseIf fVel(iPopindex, iDimindex) < -fMAXVEL Then
    fVel(iPopindex, iDimindex) = -fMAXVEL
  End If
Next iDimindex

'If it's going the right way, keep going
If iBetter(iPopindex) = 1 Then
  For iDimindex = 1 To iDIMENSIONS
    fVel(iPopindex, iDimindex) = fDumVel(iDimindex)
  Next iDimindex
End If

' Graphics Loop: Graphically plot updated positions

'Erase old position for two dimensions of one particle
iXcoord = Int((picSwarm1.ScaleWidth / 2#) * (1 + _
  10 * fPos(iPopindex, 1) / fMaxPos))
iYcoord = Int((picSwarm1.ScaleHeight / 2#) * (1 + _
  10 * fPos(iPopindex, 2) / fMaxPos))
picSwarm1.PSet (iXcoord, iYcoord), vbBlack

For iDimindex = 1 To iDIMENSIONS 'Define new positions for all dimensions
  fPos(iPopindex, iDimindex) = fPos(iPopindex, iDimindex) + _
    fVel(iPopindex, iDimindex)
Next iDimindex

'Graph new position for two dimensions of one particle
iXcoord = Int((picSwarm1.ScaleWidth / 2#) * (1 + _
  10 * fPos(iPopindex, 1) / fMaxPos))
iYcoord = Int((picSwarm1.ScaleHeight / 2#) * (1 + _
  10 * fPos(iPopindex, 2) / fMaxPos))
picSwarm1.PSet (iXcoord, iYcoord), vbRed

'If local neighborhood, graph iLbest
If iLOCAL > 0 Then
  iXcoord = Int((picSwarm1.ScaleWidth / 2#) * (1 + _
    10 * fPos(iLbest, 1) / fMaxPos))
  iYcoord = Int((picSwarm1.ScaleHeight / 2#) * (1 + _
    10 * fPos(iLbest, 2) / fMaxPos))
  picSwarm1.PSet (iXcoord, iYcoord), vbYellow
End If

```

Next iPopindex 'end velocity, position, graph loop

'Graph 2 dimensions of iGbest

```
iXcoord = Int((picSwarm1.ScaleWidth / 2#) * (1 + _
10 * fPos(iGbest, 1) / fMaxPos))
iYcoord = Int((picSwarm1.ScaleHeight / 2#) * (1 + _
10 * fPos(iGbest, 2) / fMaxPos))
picSwarm1.PSet (iXcoord, iYcoord), vbWhite
```

'Terminate on sufficiently low error

If (fPbestVal(iGbest) < fERRCUTOFF) Or nIter >= nMAXITER Then

MsgBox "Iter: " & nIter & " Err: " & fPbestVal(iGbest) ', vbYesNo

Open sOutfile For Append As #2 'psoutfile.txt

Print #2, "Iter: " & nIter, " Best val: " & fPbestVal(iGbest), Now

Print #2, "Neighd. size (0=global): " & iLOCAL, " Fnctn# " & iFNCTNO, \_  
" #Dims " & iDIMENSIONS

For iDimindex = 1 To iDIMENSIONS

Print #2, "Dim. " & iDimindex, "Pos. " & fPos(iGbest, iDimindex)

Next iDimindex

Close #2

End

'END PROGRAM

End If

Next nIter ' next nIter, end main loop

End Sub

Sub evalf6(iPopindex) ' evaluates f6 function error: fVal(iPopindex)

Dim fNum As Double, fDenom As Double, fF6val As Double

```
fNum = (Sin(Sqr((fPos(iPopindex, 1) * fPos(iPopindex, 1) + _
fPos(iPopindex, 2) * fPos(iPopindex, 2)))) ^ 2 - 0.5
```

```
fDenom = (1# + 0.001 * (fPos(iPopindex, 1) * fPos(iPopindex, 1) + _
fPos(iPopindex, 2) * fPos(iPopindex, 2))) ^ 2
```

```
fF6val = 0.5 - (fNum / fDenom)
```

```
fErrVal(iPopindex) = 1# - fF6val
```

End Sub

Sub evalf0(iPopindex) 'Evaluates spherical f0 function error

fErrVal(iPopindex) = 0#

For iDimindex = 1 To iDIMENSIONS

```
fErrValDim = (fPos(iPopindex, iDimindex)) ^ 2
```

```
fErrVal(iPopindex) = fErrVal(iPopindex) + fErrValDim
```

Next iDimindex

End Sub

Sub evalf1(iPopindex) ' Evaluates Rosenbrock f1 function error

fErrVal(iPopindex) = 0#

For iDimindex = 1 To iDIMENSIONS - 1

```
fErrValDim = 100# * (fPos(iPopindex, iDimindex + 1) - (fPos(iPopindex, iDimindex)) ^ 2 + (fPos(iPopindex, iDimindex) - 1) ^ 2
```

```
fErrVal(iPopindex) = fErrVal(iPopindex) + fErrValDim
```

Next iDimindex

End Sub

Sub evalf2(iPopindex) ' Evaluates Rastrigrin f2 function error

fErrVal(iPopindex) = 0#

For iDimindex = 1 To iDIMENSIONS

```
fErrValDim = (fPos(iPopindex, iDimindex)) ^ 2 + 10# - 10# * Cos(2# * conPI * fPos(iPopindex, iDimindex))
```

```
fErrVal(iPopindex) = fErrVal(iPopindex) + fErrValDim
```

Next iDimindex

End Sub

```

Sub evalf3(iPopindex)      ' Evaluates Griewank f3 function error
Dim fFirstDim As Double, fSecondDim As Double
Dim fFirstTot As Double, fSecondTot As Double
fErrVal(iPopindex) = 0#
fFirstTot = 0#
fSecondTot = 1#

```

```

For iDimindex = 1 To iDIMENSIONS
    fFirstDim = (fPos(iPopindex, iDimindex)) ^ 2
    fFirstTot = fFirstTot + fFirstDim
    fSecondDim = Cos(fPos(iPopindex, iDimindex) / Sqr(iDimindex))
    fSecondTot = fSecondTot * fSecondDim
Next iDimindex

```

```

fErrVal(iPopindex) = (0.00025 * fFirstTot) - fSecondTot + 1#

```

```

End Sub

```

```

Sub evalf5(iPopindex)      'Evaluates He f5 function error
fErrVal(iPopindex) = (fPos(iPopindex, 1)) ^ 2 + 2# * (fPos(iPopindex, 2)) ^ 2 -
0.3 * Cos(3# * conPI * fPos(iPopindex, 1)) - 0.4 * Cos(4# * conPI * fPos(iPopindex, 2))

```

```

+ 0.7

```

```

End Sub

```

```

Private Sub Form_Load()
frmSwarm.Show
End Sub

```

File: E:\Work\WRAIR\Data\actfile.txt 4/23/99, 12:34:28AM

---

## Actigraph data file

308act.xls	309act.xls	310act.xls
330act.xls	331act.xls	332act.xls
333act.xls	334act.xls	335act.xls
337act.xls	349act.xls	350act.xls
351act.xls	369act.xls	370act.xls
371act.xls	372act.xls	

3\*.xls 17

515act.xls	516act.xls	517act.xls
518act.xls	519act.xls	520act.xls
521act.xls	522act.xls	542act.xls
543act.xls	544act.xls	545act.xls
553act.xls	554act.xls	555act.xls
556act.xls		

5\*.xls 16

711act.xls	712act.xls	713act.xls
714act.xls	723act.xls	724act.xls
725act.xls	726act.xls	738act.xls
739act.xls	740act.xls	741act.xls
757act.xls	758act.xls	759act.xls
760act.xls		

7\*.xls 16

905act.xls	906act.xls	927act.xls
928act.xls	929act.xls	946act.xls
947act.xls	948act.xls	961act.xls
962act.xls	963act.xls	964act.xls
965act.xls	966act.xls	967act.xls
968act.xls		

9\*.xls 16

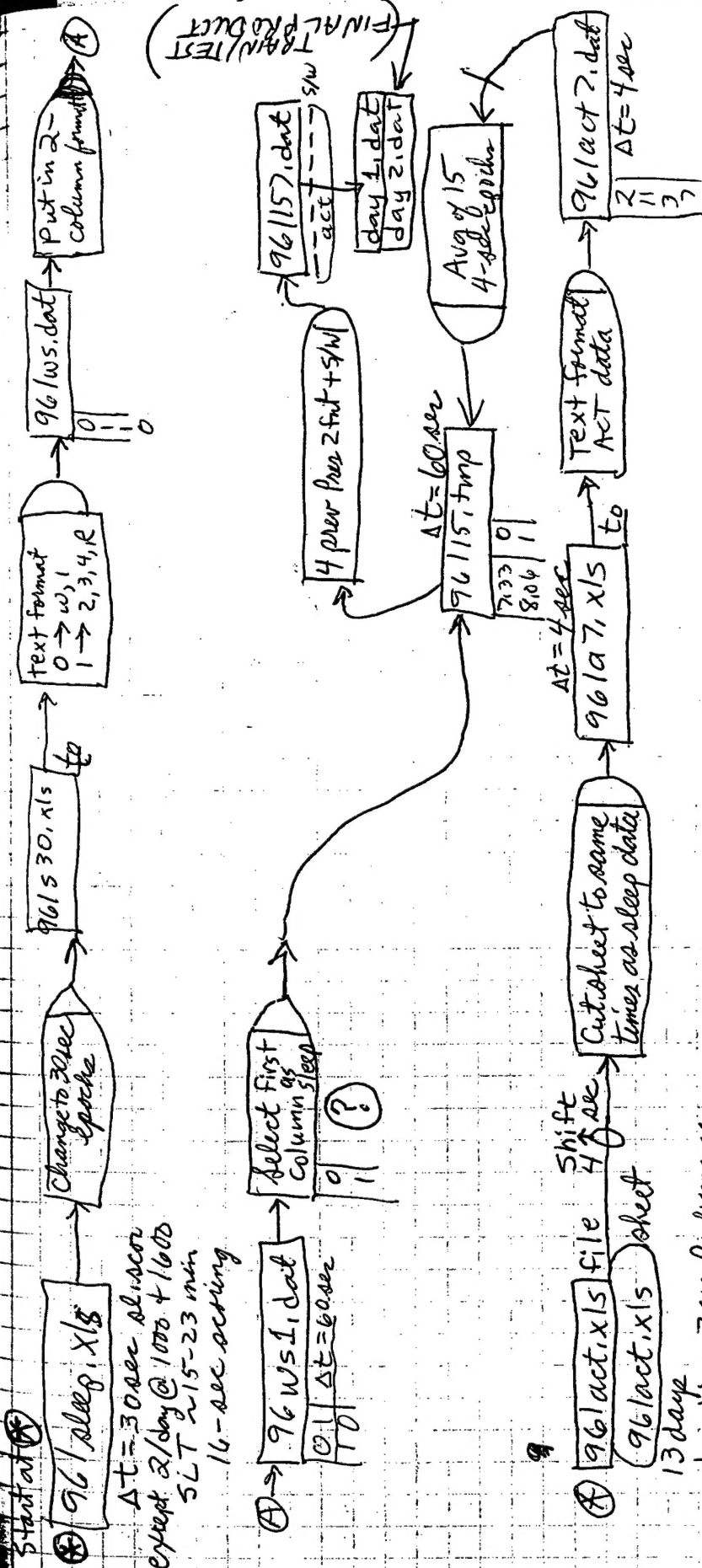
total 65 files

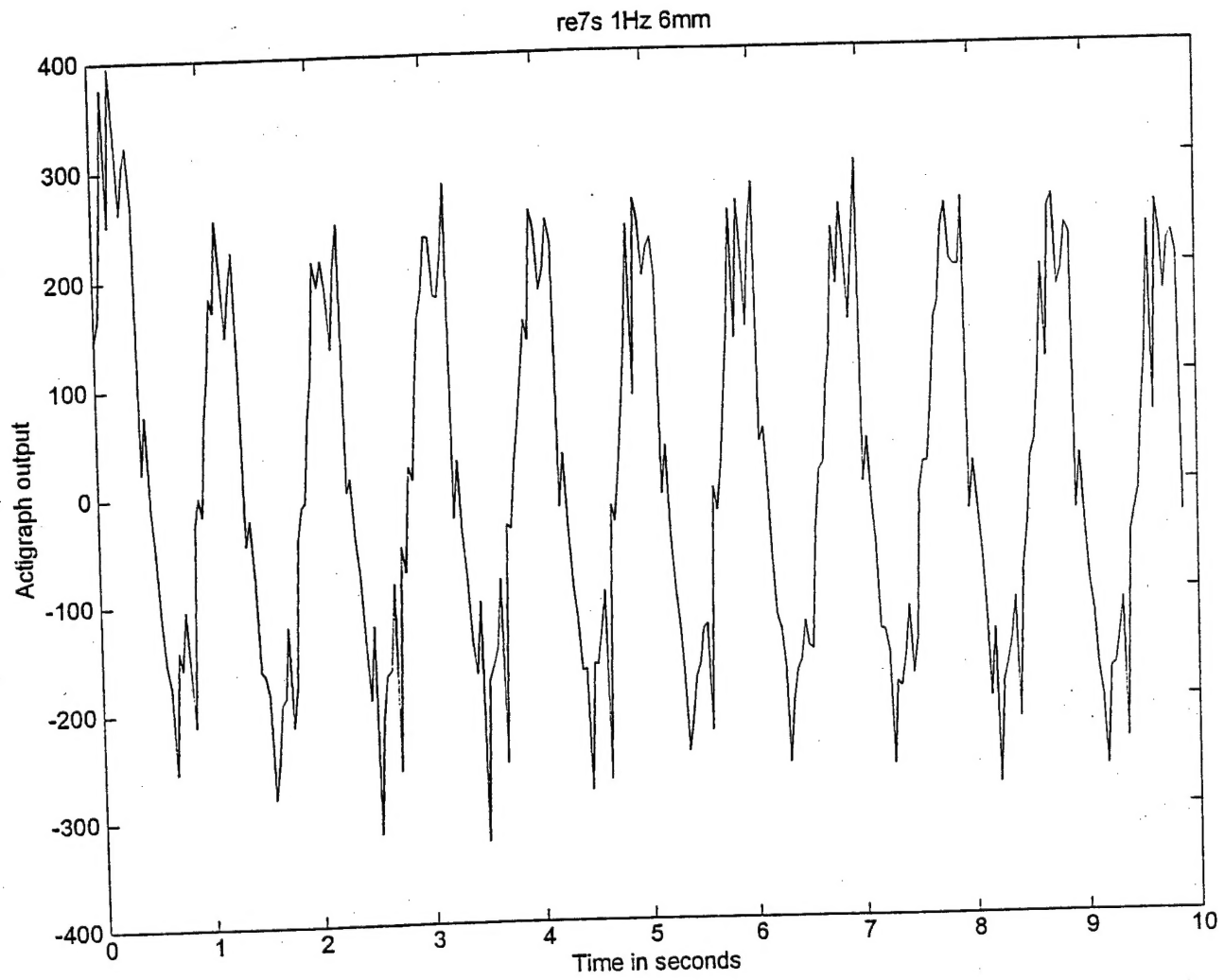
-----  
PSG data file

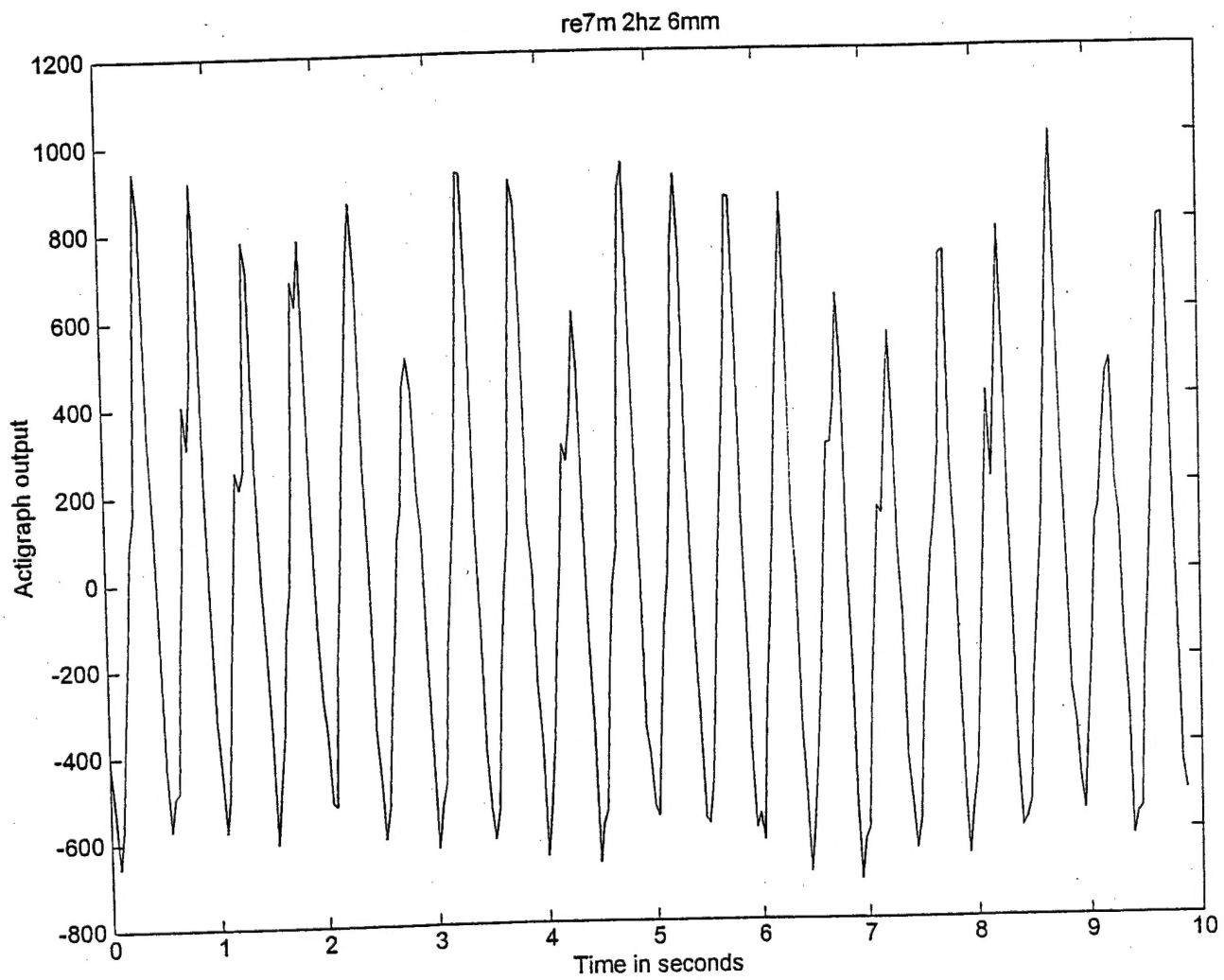
372sleep.xls  
555sleep.xls  
711sleep.xls  
961sleep.xls

total 4 files

# APPENDIX D









re7s 3Hz 6mm

